PiP: Planning-informed Trajectory Prediction for Autonomous Driving

ECCV Supplementary material

Paper ID 3805

1 Implementation Details

The PiP architecture is comprised of planning coupled module, target fusion module and maneuver-based decoding module, the detailed parameters of which are given in Table 1.

The planning coupled module operates on each target-vehicle-centric area $\mathcal{A}_{nbr}$ which is centered at predicted target $v_i \in V_{tar}$. All trajectories involved in each V_{tar} are embedded as 32-dimensional vectors by temporal convolution layer (a.k.a 1D convolution). Afterwards, the history tracks and future planning are separately encoded by LSTMs with hidden state size 64, in which the planned trajectory of ego vehicle is encoded in a reversed time order considering the near future part of planning would weight more in the sequence. The planning tensor and observation tensor are initialized as zero with 64 channels and the same shape with the discretized grid of target area 25×5, and then filled by putting trajectory encodings in the corresponding locations using binary masks. After that, the planning tensor and observation tensor pass through two branches with convolutional layers and max-pooling layer. Next, they are concatenated together before feeding to the last social pooling layer. The consequent social contextual encoding $\mathcal{S}$ of size 80 is concatenated with the target's dynamic encoding $\mathcal{D}$ obtained from the linear embedding of size 32 to form an encoding $\mathcal{T}_i$ for each target vehicle. All convolutional and linear layers in this module perform $LeakyReLU$ activation with negative slop 0.1.

The target fusion module operates on the ego-vehicle-centric area $\mathcal{A}_{tar}$ with a fully convolutional (FCN) architecture to model correlations in the predicted trajectories. The target tensor is initialized as zero with shape of $25 \times 5 \times 112$ (wideth $\times$ height $\times$ channel), and then filled by placing the target encodings $\mathcal{T}_i$ of all predicted targets into the corresponding locations using a binary mask. Sequentially, the target tensor passes through an FCN-like structure with 2 convolutional layers and 2 transposed convolutional layers. $ReLU$ activation and batch normalization are applied after each layer, and element-wise sum is performed for the skip-connection between layers. By concatenating the feature maps from the last upsampling layer with the input target tensor, the fused target tensor is produced. Finally, a fused target encoding $\mathcal{T}_i^+$ of size 224 could be sliced out for each target according to its grid location.

The maneuver-based decoding module operates on each target vehicle. Each fused target encoding $\mathcal{T}_i^+$ is fed into a pair of fully connected (fc) layers with

Table 1. Detailed parameters of PiP architecture

Stage	Key layers (kernel, output dimension)	Comment
Planning coupled module	Conv1d $(3 \times 1, 32)$	Motion embedding of all trajectories.
	LSTM (64)	Encoding planned trajectory.
	LSTM (64)	Encoding history trajectories.
	$\begin{bmatrix} \text{Conv2d } (3 \times 3, 64) \\ \text{MaxPool } (3 \times 3, \text{stride: } = 2) \\ \text{Conv2d } (3 \times 3, 16) \end{bmatrix} \times 2$	Two branches to process planning tensor and observation tensor.
	MaxPool $(2 \times 2$, stride:=2)	Pooling the merged social context.
	Linear (32)	Encoding target vehicle's dynamics.
Output: target encoding $\mathcal{T}_i$ aggregates all controlled and observed information within each $\mathcal{A}_{nbr}$		
Target fusion module	Conv2d $(3 \times 3, 224)$	Downsampling layers.
	Conv2d $(3 \times 3, 448)$	
	ConvTranspose $(3 \times 3, 224$, stride:=2)	Upsampling layers.
	ConvTranspose $(3 \times 3, 112$, stride:=2)	
Output: fused target encoding $\mathcal{T}_i^+$ is sliced out from the fused target tensor formed within $\mathcal{A}_{tar}$.		
Maneuver based decoding	Linear (2)	Probabilities of 2 longitudinal behaviors.
	Linear (3)	Probabilities of 3 lateral behaviors.
	LSTM (128)	Decoding future time frames.
	Linear (5)	Producing parameters of Gaussian distributions.
Output: Gaussian distributions of prediction under vehicle maneuver m_k and probability $P(m_k)$.		

softmax activation to generate the probabilities of 2 longitudinal behaviors and 3 lateral behaviors, respectively. The multiplication of the longitudinal probability and lateral probability produces the vehicle maneuver probability $P(m_k|k = 1, 2, ..., 6)$. The trajectory decoder is implemented by an LSTM decoder with hidden state size 128, followed by a *fc* layer that produces 5-dimensional parameters of Gaussian distributions at every future time frame. The predicted trajectory under each kind of vehicle maneuver is consequently generated from feeding the concatenation of fused target encoding $\mathcal{T}_i^+$, a one-hot vector of longitudinal maneuver and a one-hot vector of lateral maneuver to the decoder.

The training process adopts Adam optimizer with an initial learning rate of 0.001 to update the network iteratively with a batch size of 64. The network is implemented on PyTorch.

2 Runtime Analysis

The PiP architecture contains 2.58M parameters in total, with 92Hz inference frequency when running on NVIDIA GTX 1080-Ti GPU. Here it should be emphasized that different from stochastic prediction models that generate more diverse results with more runs, PiP is deterministic that only forward propagates once to predict trajectories under different maneuvers together with maneuver probabilities. In practical use, all candidate plans of ego vehicle could be assembled in a batch before feeding to the network. PiP operates at 29Hz when predicting for a batch of 20 candidate plans, which could meet the real-time requirement of most planning and tracking modules in autonomous vehicles.

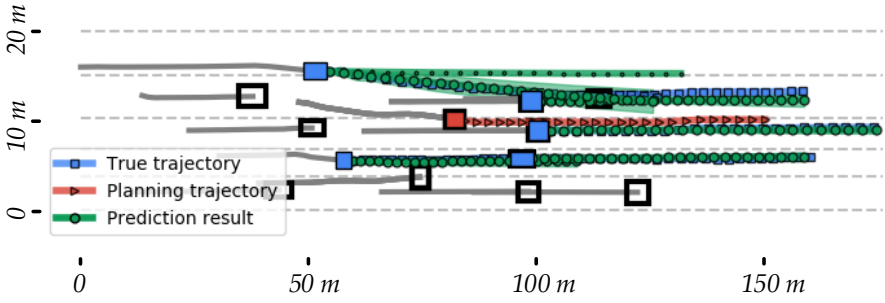

Fig. 1. Example frame in the sequential prediction results.

3 Video Description

Please view the video attachment for more prediction results. All the results are generated from running PiP on the test sets of NGSIM and HighD.

For sequential prediction results, we adopt the coloring scheme shown in Fig. 1. The lane boundaries are depicted by grey dashed lines. Red, blue and white rectangles respectively represent the ego vehicle, target vehicles and other surrounding vehicles. The true future trajectories (blue), actual planning trajectory (red) and predicted future trajectories (green) are of 5s time length and visualized by sets of waypoints with 0.2s time step, while all the historical trajectories are of 3s time length and shown by grey curves. For the reason that our model generates maneuver-based multi-modal prediction results, each target vehicle has prediction results corresponding to different vehicle maneuvers, in which only the predicted trajectories with vehicle maneuver probability $P(m_k)$ larger than 10% are visualized. We use the green circles to denote the mean values of the predicted distribution on each future time step, and the radius of the green circle is proportional to $P(m_k)$ under the corresponding trajectory. Besides, the variances of the predicted distribution are represented by the green shadow areas.

For demonstrating the user study and the active planning results, we display the trajectory animation using the mean values of the predicted distributions as waypoints, without showing the distributions.